WE CLAIM:

1. A method of transporting a concatenated input signal across a network using signals transmitted over a hyper-concatenated connection between a start node and an end node in the network, the method comprising steps of:

Sub B1

   a) receiving the concatenated input signal at the start node and splitting the concatenated input signal into a plurality of derived signals;

   b) transmitting the derived signals over a predetermined number of independent channels, at least one of the independent channels being routed through a pointer processing state machine that is independent of a pointer processing state machine through which another one of the independent channels are routed; and

   c) recombining the derived signals at the end node to form a concatenated output signal equivalent to the concatenated input signal.

2. The method as claimed in claim 1 wherein the concatenated output signal is output from the end node at a signal phase that is arbitrarily related to a signal phase of the derived signals.

3. A method as claimed in claim 1, wherein the independent channels in the hyper-concatenated connection meet predetermined criteria, comprising:

a) each of the channels are processed by adjacent pointer processors in the start node and the end node;

b) the channel order is identical at the start node and the end node; and

c) a maximum latency between the derived signals received at the end node on channels of the hyper-concatenated connection is less than a predetermined time interval.

4. A method as claimed in claim 3, wherein the predetermined interval is less than a time period required to receive a frame from a one of the derived signals at the end node.

5. A method as claimed in claim 1, wherein the concatenated input signal comprises an arbitrary mix of concatenated and unconcatenated Synchronous Optical Network (SONET)/Synchronous Digital Hierarchy (SDH) signals.

6. A method as claimed in claim 5 wherein the step of splitting the concatenated input signal into the plurality of derived signals comprises a step of inspecting an overhead of each frame of the concatenated input signal to determine whether the overhead contains a payload pointer or a concatenation indicator.

7. A method as claimed in claim 6 wherein the method further comprises a step of storing the payload

pointer if a payload pointer is found in the overhead.

8. A method as claimed in claim 7 further comprising a step of setting SS bits of an H1 byte of the overhead portion of the frame to a default value.

9. A method as claimed in claim 8, wherein the default value is binary "00".

10. A method as claimed in claim 8 further comprising steps of:

a) comparing a frame count with a predetermined constant to determine if the frame should be transmitted over a next independent channel in the hyper-concatenated connection;

b) if the frame is to be transmitted over the next independent channel, examining the frame overhead to determine whether it contains a concatenation indicator; and

c) if the frame overhead contains a concatenation indicator, replacing the concatenation indicator with the stored payload pointer, and inserting a split indicator into the SS bits.

11. A method as claimed in claim 10 further comprising a step of incrementing the frame count by one if the frame count is not equal to the predetermined constant.

12. A method as claimed in claim 10 wherein if the frame is to be transported over a next independent channel, the method further comprises steps of:

    a) incrementing a channel index by one unless the channel index equals a number of channels in the hyper-concatenated connection;

    b) setting the channel index to one if the channel index equals the number of channels in the hyper-concatenated connection; and

    c) transmitting the frame over the independent channel identified by the channel index.

13. The method as claimed in claim 1 wherein the step of recombining the derived signals at the end node to form a concatenated output signal further comprises steps of:

    a) examining an overhead of each frame of the derived signals to determine whether the overhead includes a split indicator;

    b) if the overhead includes a split indicator, replacing a payload pointer in the overhead with a concatenation indicator; and

    c) reading out payload data of the derived signals in alignment across all of the channels of the hyper-concatenated connection to provide the concatenated output signal.

14. A method as claimed in claim 13, wherein the step of reading out the payload data of the derived signals received at the end node comprises steps of:

a) designating a data steam in the hyper-concatenated connection as a reference data signal;

b) designating all other data steams of the hyper-concatenated connection as slaves to the reference data stream;

c) controlling a read operation for reading the payload data of the reference data stream from a reference alignment buffer such that a position of a reference read pointer permits corresponding payload data to be read simultaneously from each slave data stream; and

d) reading the payload data of each slave data stream from respective slave alignment buffers based on the reference read operation, so that payload data of each of the slave data streams is read from the respective slave alignment buffers in alignment with corresponding payload data of the reference data stream.

15. A method as claimed in claim 14 wherein the alignment buffers have a predetermined storage capacity based on an anticipated maximum difference between propagation delays of the respective derived signals.

16. A method as claimed in claim 15, wherein the anticipated maximum difference between propagation delays of the respective derived signals is less than a time interval required to receive one of the frames at the end node.

17. A method as claimed in claim 15, wherein the predetermined storage capacity is sufficient to store a quantity of payload data received in approximately twice the anticipated maximum difference between propagation delays of the respective frames of each derived signal.

18. A method as claimed in claim 17, wherein the predetermined storage capacity is adequate to store a quantity of payload data received during a time interval of about 250 μSec.

19. A network node adapted to function as a start node for a hyper-concatenated connection across a network between the start node and an end node, the hyper-concatenated connection being routed through at least one independent pointer processing state machine, the network node comprising:

a) an input port adapted to receive a concatenated input signal;

b) a signal processor adapted to inverse-multiplex the concatenated input signal across a plurality of channels of the hyper-concatenated connection; and

c) an output port adapted to launch the inverse-multiplexed concatenated input signals across the network as hyper-concatenated data streams within respective ones of the channels.

20. A network node as claimed in claim 19, wherein each hyper-concatenated channel has a signal bandwidth

expressed as an integer M (where $M \geq 1$) of frames of the demultiplexed input signal to be transmitted over each respective channel.

21. A network node as claimed in claim 20, wherein M is selected from a group consisting of: 1, 2, or an integer multiple of 3.

22. A network node as claimed in claim 19, wherein the concatenated input signal comprises an arbitrary mix of concatenated and unconcatenated Synchronous Optical Network (SONET)/Synchronous Digital Hierarchy (SDH) signals.

23. A network node as claimed in claim 22, wherein each frame is an STS-n, where n is an integer, and $n \geq 1$.

24. A network node as claimed in claim 23, wherein n is equal to 1.

25. A network node as claimed in claim 20, wherein the signal processor comprises:

a) means for identifying each frame within the concatenated input signal that satisfies a condition (p*M)+1, for integers p, in order to determine a split location for the concatenated input signal;

b) means for modifying each frame that satisfies the condition if an overhead of the respective frames includes a concatenation indicator; and

c) means for determining a channel for transporting a frame that satisfies the condition, and each

subsequent frame until a next frame satisfies the condition.

26. A network node as claimed in claim 25 wherein the means for modifying the frames that satisfy the condition is a pointer processor.

27. A network node as claimed in claim 26, wherein the pointer processor is adapted to modify the frames having an overhead that includes a concatenation indicator by:

a) inserting a split indicator into a predetermined location within the frame overhead; and

b) replacing the concatenation indicator in the frame overhead with a stored payload pointer extracted from an overhead of a first frame of a concatenated signal that includes the frame meeting the condition.

28. A network node as claimed in claim 21, wherein the signal processor is adapted to determine the split in the concatenated input signal in real-time as the concatenated input signal is received by the network node.

29. A network node as claimed in claim 27, wherein the split indicator is inserted by assigning a predetermined value to SS bits of an H1 byte of an overhead portion of the frame.

30. A network node as claimed in claim 29, wherein the predetermined value is binary "01".

31. A network node adapted to function as an end node for a hyper-concatenated connection between a start node and the end node, the hyper-concatenated connection being routed through independent pointer processing state machines, the network node comprising:

    a) an input port adapted to receive hyper-concatenated data streams from adjacent channels;

    b) a signal processor adapted to combine the hyper-concatenated data streams into a concatenated output signal; and

    c) an output port adapted to transmit the concatenated output signal to a downstream node.

32. A network node as claimed in claim 31, wherein the concatenated output signal comprises an arbitrary mix of concatenated and unconcatenated Synchronous Optical Network (SONET)/Synchronous Digital Hierarchy (SDH) signals.

33. A network node as claimed in claim 31, wherein the signal processor comprises, in respect of each hyper-concatenated data stream:

    a) an alignment buffer adapted to buffer payload data of a respective hyper-concatenated data stream;

    b) a pointer processor adapted to detect a frame received in a respective data stream and determine a location of payload data in the frame;

c) a read controller responsive to the pointer processor and adapted to read the buffered payload data in an aligned condition across the channels of the hyper-concatenated connection into the concatenated output signal.

34. A network node as claimed in claim 33, wherein the pointer processor further comprises:

a) a signal monitor adapted to monitor at least an overhead portion of the respective frames; and

b) means for determining whether the overhead portion contains a split indicator.

35. A network node as claimed in claim 33, further comprising a control means adapted to:

a) designate one of the hyper-concatenated data steams as a reference data stream; and

b) designate all others of the hyper-concatenated data steams as slave data streams.

36. A network node as claimed in claim 35, wherein a reference read controller is adapted to control a reference read operation for reading payload data of the reference data stream from a respective reference alignment buffer so that payload data from each of the slave data streams can be read by respective slave read operations in alignment with the reference data stream.

37. A network node as claimed in claim 33, wherein the read controller is adapted to replace a payload

pointer in a received frame with a concatenation indicator if an overhead of the received frame contains a split indicator.

38. A network node as claimed in claim 33, wherein the alignment buffer has a predetermined storage capacity based on an anticipated maximum difference between propagation times of the respective signals received on each hyper-concatenated data stream.

39. A network node as claimed in claim 38, wherein the predetermined storage capacity of the alignment buffer is adequate to store frame data received in a time interval equivalent to approximately twice the anticipated maximum difference in propagation delay of the respective hyper-concatenated data streams.

40. A network node as claimed in claim 38, wherein the anticipated maximum difference in propagation delay between the respective hyper-concatenated data steams is less than a time interval required to receive a frame at the end node on any one of the hyper-concatenated data streams.

41. A network node as claimed in claim 38, wherein the predetermined storage capacity is adequate to store frame data received during a period of approximately 250 µSec.